REMARKS

The claims previously extant in the application have been reviewed and amended, where appropriate, for clarity. The specification has been amended to correct a few errors of a clerical nature. Claims 12-20 have been added to provide Applicants to the protection to which they are deemed entitled.

Claims 8 and 9 are now directed to a program or medium carrying computer readable data for instructing a processor to perform SCSI operations. It is the understanding of attorney for applicant the PTO is now taking the position that programs, *per se*, are not patentable subject matter within the confines of 35 USC §101. The Office agrees, however, that a memory or medium carrying computer readable data for instructing a processor to perform certain operations is patentable subject matter within the confines of 35 USC §101. As a result, claims 8 and 9 are amended as stated to advance prosecution.

Applicants traverse the rejection of claims 1-11 as being obvious under 35 USC §103(a) as a result of Hausauer et al. (U.S. Patent 5,790,870).

Hausauer et al. includes a primary peripheral component interconnect (PCI) bus 117 including lead or line 190 carrying a primary channel address parity error signal (CSERR #) and lead or line 192 carrying primary channel data parity error signal (CPERR #). In addition, Hausauer et al. includes secondary PCI bus 115 including lead or line 194 carrying secondary channel address parity error signal (SSERR #) and lead or line 196 carrying secondary channel data parity error signal (SPERR #). The signal on lead or line 194 is supplied to an enable terminal of tristate enable buffer 200, while the signal on lead or line 196 is supplied to one input terminal of AND gate 202. The output of buffer 200 is connected to lead or line 190 so that when signal SSERR# on line 194 is asserted, the CSERR# signal line 190 is also asserted via buffer 200. The signals on lines or leads 192 and 196 are supplied to input terminals of AND gate 202 to enable a combined PERR# signal to be supplied to a PCI clock before presenting the PERR# signal to a second input of an interrupted controller 124, Figure 1. As indicated in column 5, lines 32-34, interrupt controller 124 responds to its input signals to route interrupt requests from devices located on PCI buses 115 and 117 to processors on slots 100 and 102.

The Examiner relies on a rather insignificant portion of Hausauer et al., dealing with SCSI controller or interface 116, Figures 1 and 3. In the discussion of SCSI controller 116 in column 5, lines 18-25, Hausauer et al. indicates SCSI controller 116 provides the capability of handling simultaneous disk commands and that SCSI 116 is connected to plural SCSI connectors 116 which drive plural disk drives adapted to support the simultaneous issuance of multiple commands by a host system to one or more SCSI devices. Column 8, lines 18-21, indicates SCSI interface device 116 has an output signal connected to the SSERR# line 194 and another output signal connected to the SPERR# line 196 of secondary PCI bus 115. The Office Action has a discussion of SCSI interface device 116 in connection with column 7, lines 44-46. However, this portion of Hausauer et al. appears to have nothing to do with SCSI interface device 116. In particular, the sentence beginning in column 7, line 44 and ending in line 46, is:

> The recommended route is for the master to inform its device driver of the error by generating an interrupt among other options.

The master in the above-quoted portion of Hausauer et al. appears to be the device which supplies PCI buses 115 and 117. The interrupted generator appears to be interrupt controller 124 that controls the processors on slots 100 and 102.

There does not appear to be any disclosure in Hausauer et al. of SCSI controller 116 or SCSI interface 116 (Figure 1 and 3) responding to the parity error signals on lines 194 and 196 of secondary PCI bus 115.

The Office Action states that column 7, lines 38-46 of Hausauer et al. discloses determining whether SCSI interface 116 is an enabled device in a data transfer state. However, there is no mention of SCSI interface 116 or SCSI controller 116 in this portion of Hausauer et al. The Examiner, if he repeats this position, is requested to specify with greater particularity how column 7, lines 38-46, includes the determining feature of claim 1.

Because Hausauer et al. does not appear to disclose determining whether SCSI interface 116 is an enabled device in a data transfer state, Hausauer et al. does not disclose or make obvious generating a response message to an initiating device if the SCSI enable device is in the data transfer state, as claim 1 requires. Consequently, Hausauer et al. does not disclose or make obvious that such a generated response message notifies an initiator device that a previous data transfer operation should be recommenced, as claim 1 also requires.

The Examiner's reliance on column 7, lines 42-46 and column 9, lines 59-64 of Hausauer et al. to make the foregoing features of claim 1 obvious is misplaced. There is no mention of SCSI interface or controller 116 in either of these portions of Hausauer et al. Column 9, lines 57-64, indicates that in step 408, the processor polls each device on the primary and secondary PCI buses that is capable of asserting the error for the particular non-maskable interrupt (NMI) error group. There is no indication that SCSI interface 116 is in the NMI error group. Several NMI error groups are listed in the table in column 9, lines 26-35; none of the entries refer to SCSI interface 116 or the tape drives coupled to it.

Based on the foregoing, the rejection of claim 1 as being obvious as a result of Hausauer et al. is incorrect. Claim 5, directed to a SCSI enabled device is also improperly rejected for the same rationale as discussed in connection with claim 1. Further, claim 10, also directed to a SCSI enabled device, is patentable over Hausauer et al. for the same reasons advanced with regard to claim 1.

The allegation in the Office Action that Hausauer et al. discloses the claim 2 requirement for recommencing the data transfer operation from the start of the data transfer operation is incorrect. The Examiner relies on column 9, lines 61-64, of Hausauer et al. for this feature. However, the sentence beginning in line 61 and ending in line 64 of col. 9 states:

> The corrective action may be as simple as informing the user an error has occurred or may be as complex as requiring certain software to be re-executed and data transmitted.

Re-execution of certain software and retransmission of data is not necessarily the same thing as recommencing a data transfer operation from the start of the data transfer operation.

Concerning independent claim 3, Hausauer et al. does not disclose that SCSI interface device 116 receives a message parity error message following a data transfer phase of SCSI interface device 116. The reliance on column 7, lines 33 and 34, as well as 59 and 60, is misplaced. There is no mention whatsoever in these two portions of

Hausauer et al. of SCSI interface 116. Further, as admitted by the Office Action, Hausauer et al. does not indicate that SCSI interface 116 sends a restore data pointer message after SCSI interface 116 receives a message parity error message, as required by claim 3. The reliance by the Office Action on column 7, lines 42-46, and column 9, lines 59-64, of Hausauer et al. for making this feature obvious is incorrect. There is nothing in these portions of Hausauer et al. to indicate or lead one of ordinary skill in the art to conclude that SCSI interface 116 sends a restore data pointer message in the manner required by claim 3.

The reliance in the Office Action on Hausauer et al., column 9, lines 61-64, for the requirement of claim 4 for the SCSI interface 116 to send a message instructing recommencing of the data transfer phase after sending the restore data pointer message is incorrect. The relied upon portion of Hausauer et al. fails to have anything to do with SCSI interface 116, as discussed previously.

Claim 6 is allowable for the same reasons advanced for claim 3 because claim 6 is directed to a SCSI driver that is capable of performing the operations set forth in claim 3.

The rejection of claim 3 as being obvious as a result of Hausauer et al. is incorrect for the same reasons advanced for claim 4.

Claim 9 is allowable for the same reasons advanced for claim 8, upon which claim 9 depends.

Claim 10 is allowable for the same reasons advanced for claim 1 and claim 11 is allowable for the same reasons advanced for claim 6.

Newly added claim 12 is directed to a method of avoiding a possible crash or hang in a peripheral device caused under conditions in which a host computer device seizes a SCSI bus during a bus fill period after an arbitration host selection period and the commencement of operation of the peripheral device. The method includes enabling the peripheral device via a SCSI bus. While the peripheral device is so enabled, activating a driver coupled with the peripheral device via the SCSI bus, so the driver supplies the peripheral device with a signal sequence on the SCSI bus. The signal sequence normally includes a data transfer phase during which the data are transferred between the host computer device and the peripheral device, followed by a message phase that includes a message parity error message. The response signal is generated by the peripheral device in response to receipt thereby of the message parity error message. The driver performs the following steps (a), (b) and (c) in response to a message parity error message being on a SCSI bus to which the driver is responsive. The message parity error occurs immediately after the data transfer. The steps are (a) determining if the driver is in the data transfer phase, (b) if the driver is not in the data transfer phase, causing the driver to continue to respond to the message parity error message in a conventional manner, (c) if the driver is in the data transfer phase, causing the driver to recognize the message parity error message as being a SCSI non-operation message. The peripheral device responds to step (c) of the driver operation by sending a restore data pointer message back to the computer device. The restore data pointer message informs that the computer peripheral device is going to re-try the entire data transfer phase from the beginning.

Claim 13 depends on claim 12 and requires the data transfer to be resumed at the computer device from the beginning in response to receipt at the computer device of the restore data pointer message. Claim 14 requires the peripheral device to resume the data transfer from the beginning in response to the computer device resuming the data transfer from the beginning. Claims 12-14 clearly distinguish over the art of record.

Claims 15-17 are apparatus for performing the method of claims 12-14, while claims 18-20 are directed to a memory or a medium including machine readable indicia for causing a computer system to execute the method of claims 12-14. Claims 12-18 are allowable for the reasons advanced for the claims upon which they depend.

In view of the foregoing amendments and remarks, favorable reconsideration and allowance are respectfully requested and deemed in order.

To the extent necessary, a petition for an extension of time under 37 C.F.R. 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 08-2025 and please credit any excess fees to such deposit account.

Respectfully submitted,
Simon PELLY et al.

By: Allan M. Lowe
Allan M. Lowe
Reg. No. 19,641

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, CO 80527-2400
703-684-1111 telephone
970-898-0640 telecopier
AML:alw

M:\1509 Hewlett-Packard Company\1509-273\Amend 11-12-04 OA.doc

CERTIFICATION OF FACSIMILE TRANSMISSION
I HEREBY CERTIFY THAT THIS PAPER IS BEING FACSIMILE TRANSMITTED TO THE PATENT AND TRADEMARK OFFICE ON THE DATE SHOWN BELOW

Roseanna Karlan
TYPE OR PRINT NAME OF PERSON SIGNING CERTIFICATION

3-14-05
SIGNATURE DATE

703-872-9306
FACSIMILE NUMBER